

マイコンカーラリー応用キット

SPI 接続超小型グラフィック液晶 (ST7565R 使用) プログラム解説 マニュアル (R8C/38A 版)

本マニュアルで説明しているセット内容	(株)秋月電子通商 製の「超小型グラフィックLCDピッチ変換キット 通販コード:K-07007」を制御します。
本基板の対象マイコンボード	RY_R8C38 ボード
本基板の製作(結線)についての説明	本マニュアルで解説
本基板のプログラムについての説明	本マニュアルで解説

第 1.01 版

2015.04.20

株式会社日立ドキュメントソリューションズ

注意事項 (rev.6.0H)

著作権

- ・本マニュアルに関する著作権は株式会社日立ドキュメントソリューションズに帰属します。
- ・本マニュアルは著作権法および、国際著作権条約により保護されています。

禁止事項

ユーザーは以下の内容を行うことはできません。

- ・第三者に対して、本マニュアルを販売、販売を目的とした宣伝、使用、営業、複製などを行うこと
- ・第三者に対して、本マニュアルの使用権を譲渡または再承諾すること
- ・本マニュアルの一部または全部を改変、除去すること
- ・本マニュアルを無許可で翻訳すること
- ・本マニュアルの内容を使用しての、人命や人体に危害を及ぼす恐れのある用途での使用

転載、複製

本マニュアルの転載、複製については、文書による株式会社日立ドキュメントソリューションズの事前の承諾が必要です。

責任の制限

本マニュアルに記載した情報は、正確を期すため、慎重に制作したのですが万一本マニュアルの記述誤りに起因する損害が生じた場合でも、株式会社日立ドキュメントソリューションズはその責任を負いません。

その他

- ・本マニュアルに記載の情報は本マニュアル発行時点のものであり、株式会社日立ドキュメントソリューションズは、予告なしに、本マニュアルに記載した情報または仕様を変更することがあります。製作に当たりましては、最新の内容を確認いただきますようお願いいたします。
- ・すべての商標および登録商標は、それぞれの所有者に帰属します。

連絡先

株式会社 日立ドキュメントソリューションズ
〒135-0016 東京都江東区東陽六丁目3番2号 イースト21タワー
E-mail:himdx.m-carrally.dd@hitachi.com

目次

1. 概要.....	1
1.1 特徴.....	1
1.2 購入先.....	2
2. 仕様.....	3
3. 接続.....	4
3.1 超小型グラフィックLCDピッチ変換キットにフラットケーブルを取り付ける.....	4
3.2 マイコンボードと取り付ける.....	6
4. プログラム.....	8
4.1 ワークスペースのインストール.....	8
4.2 プロジェクト.....	10
4.3 「lcd_st7565r_lib.c」ファイル.....	11
4.3.1 使用できる関数.....	11
4.3.2 グラフィック液晶と接続するポートの変更.....	15
4.4 「spi_glcd_38a.c」ファイル.....	16
4.4.1 プログラム.....	16
4.4.2 プログラムの解説.....	18
5. 参考文献.....	20

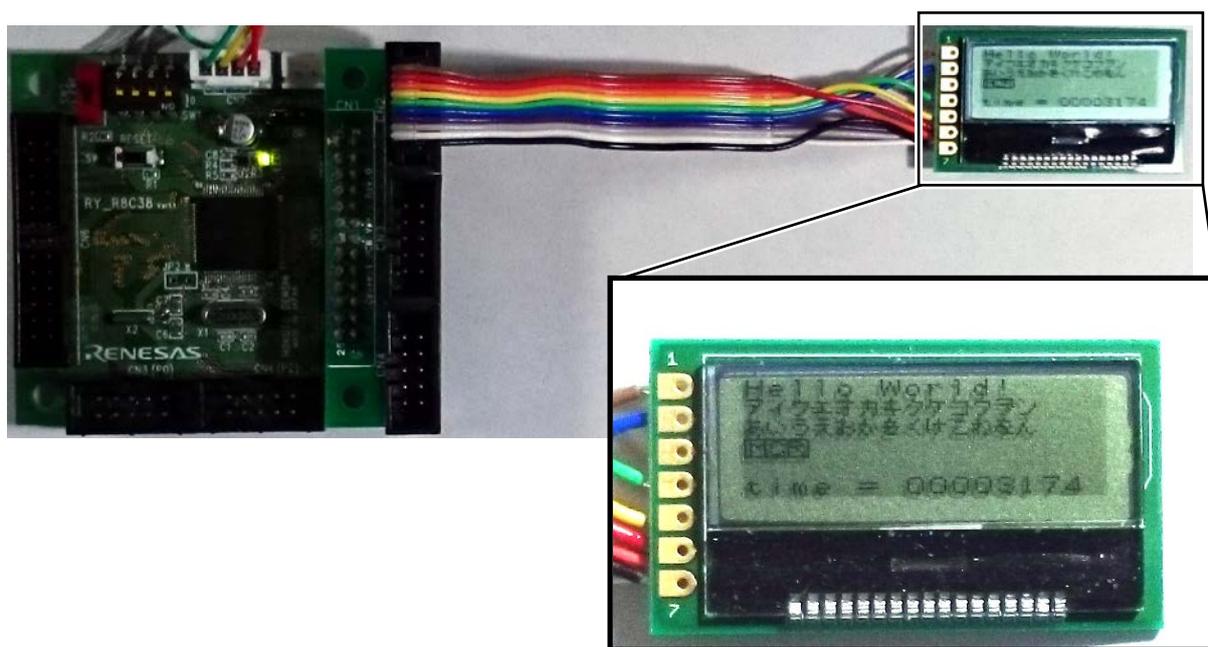
1. 概要

1.1 特徴

本マニュアルは、(株)秋月電子通商で販売されている「超小型グラフィックLCDピッチ変換キット 通販コード：K-07007」(以下、グラフィック液晶)を、RY_R8C38 ボード (R8C/38A マイコン) で制御するための結線、プログラムについて解説しています。

本基板の特徴を、下記に示します。

- グラフィック液晶との結線は、+5V、GND、SDI、SCLK、RS、CS の各端子の 6 本のみです。
- R8C/38A マイコンの「チップセレクト付きクロック同期形シリアルI/O (SSU)」機能を使用することにより、プログラムの負荷を減らして制御することができます(ただしマイコンのポートとグラフィック液晶との接続は、P3_7 端子と SDI 端子、P3_5 端子と SCLK 端子を接続する必要があります。RS 端子、CS 端子は変更可能です)。 ※具体的には、割り込みプログラムで実行している lcdShowProcess 関数の処理時間が、液晶・microSD 基板の液晶より短くなります。
- R8C/38A マイコンの P3_7 端子、P3_5 端子以外にも接続可能 (SSU は使いません)。ただし、プログラムの負荷が増えます。 ※具体的には、割り込みプログラムで実行している lcdShowProcess 関数の処理時間が、液晶・microSD 基板の液晶より長くなります。



▲RY_R8C38 ボードに RY_R8C38 ボードコネクタ変換基板を取り付け、グラフィック液晶を接続、表示しているところ

1.2 購入先

品名	購入先	写真
<p>RX_R8C38 ボード</p>	<p>(株)日立ドキュメントソリューションズの販売 サイト https://www2.himdx.net/mcr/ 型式:M-S181</p> <p>※コネクタは各自で実装、写真は半田付 け例</p>	
<p>RX_R8C38 ボードコネクタ変換基板</p> <p>※無くても結線は 可能です</p>	<p>(株)日立ドキュメントソリューションズの販売 サイト https://www2.himdx.net/mcr/ 型式:M-S185</p> <p>※本製品はキット、写真は完成例</p>	
<p>超小型グラフィックLCD ピッチ変換キット</p>	<p>(株)秋月電子通商 http://akizukidenshi.com/catalog/g/gK-07007/ 型式:K-07007</p> <p>※本製品はキット、写真は完成例</p>	

2. 仕様

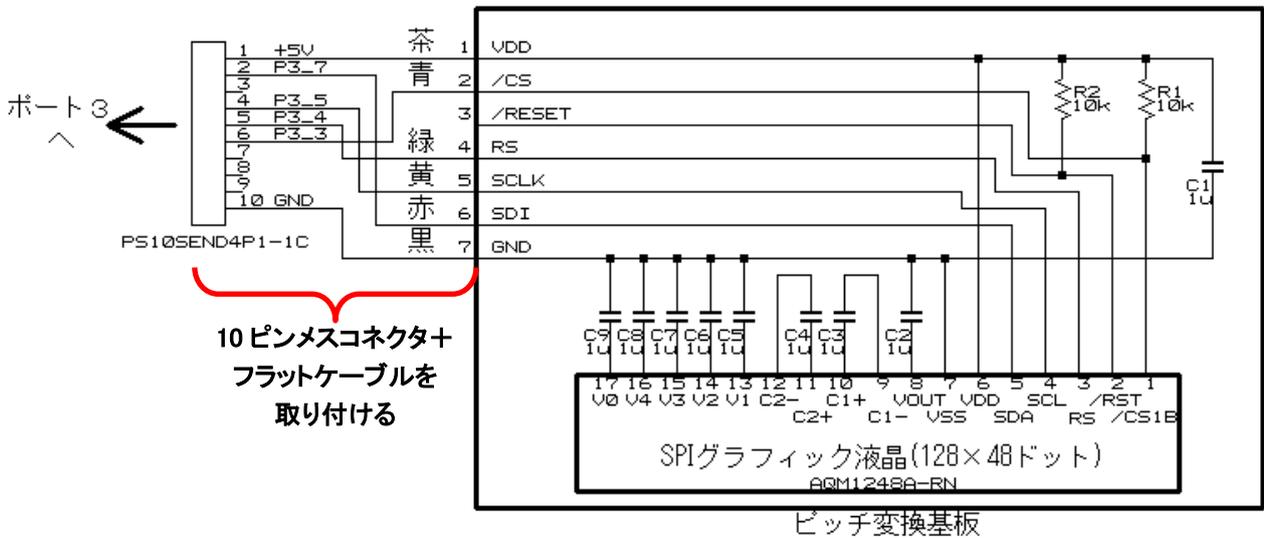
グラフィック液晶の仕様を下記に示します。参考に、液晶・microSD 基板の液晶も掲載しておきます。

	超小型グラフィック液晶	液晶・microSD 基板 (参考)
液晶	秋月電子通商製 超小型グラフィックLCDモジュール 1文字8×8ドットで表示させると、16×6 文字の表示が可能	秋月電子通商製 LCDキャラクタディスプレイモジュール(1 6×2行バックライト無)
写真 (液晶を基板に半田 付けした完成例)	 <p>※秋月電子通商製の超小型グラフィックLCDモジュールを超小型グラフィックLCDピッチ変換キットに取り付けた例</p>	 <p>※日立ドキュメントソリューションズ製の液晶・microSD 基板に取り付けた例</p>
マイコンとの接続線	4本(SDI 端子、SCLK 端子、 RS 端子、CS 端子)	7本(E 端子、WR 端子、RS 端子、 D3 端、D2 端子、D1 端子、D0 端子)
液晶のサイズ (基板は含みません)	34×20×厚さ約 1mm	85×30×厚さ 13mm
液晶の重さ (基板は含みません)	約 1g	約 32g
表示数	横 16 文字、縦 6 行 ※1文字8×8ドットで表示させたとき	横 16 文字、縦 2 行
濃さの調整	プログラムで設定	外付けで取り付ける 10kΩ のボリュームで調整
特徴	<ul style="list-style-type: none"> •小さく軽い •文字以外のグラフィックを表示することができる •文字もグラフィックで表示させないといけないので、フォントデータが必要 (ROM 容量が多くなる) 	<ul style="list-style-type: none"> •グラフィック液晶よりは大きく重たい •表示文字は 32 文字(16×2)

3. 接続

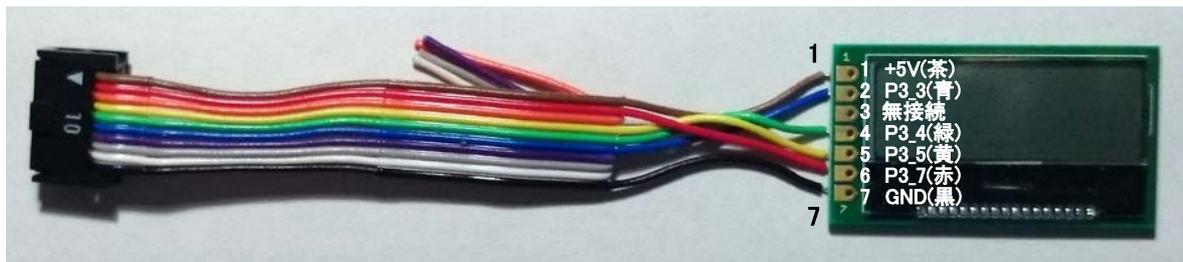
3.1 超小型グラフィックLCDピッチ変換キットにフラットケーブルを取り付ける

- 変換キット付属のピッチ変換基板に、グラフィック液晶を半田付けします。
- 基板には7個のランドがあります。回路図のように6本の線(フラットケーブルなど)でピッチ変換基板と結線します(RESET 端子は未接続です)。



※できる限り、SDI 端子と SCLK 端子の接続ポートは、P3_5 と P3_7 にしてください。プログラムの負荷を軽減することができます。それ以外のポートにも接続可能ですが、プログラムの変更が必要です。RS 端子と CS 端子は、どのポートにも接続することができます。プログラムの変更については「4.3.2

グラフィック液晶と接続するポートの変更」を参照してください。

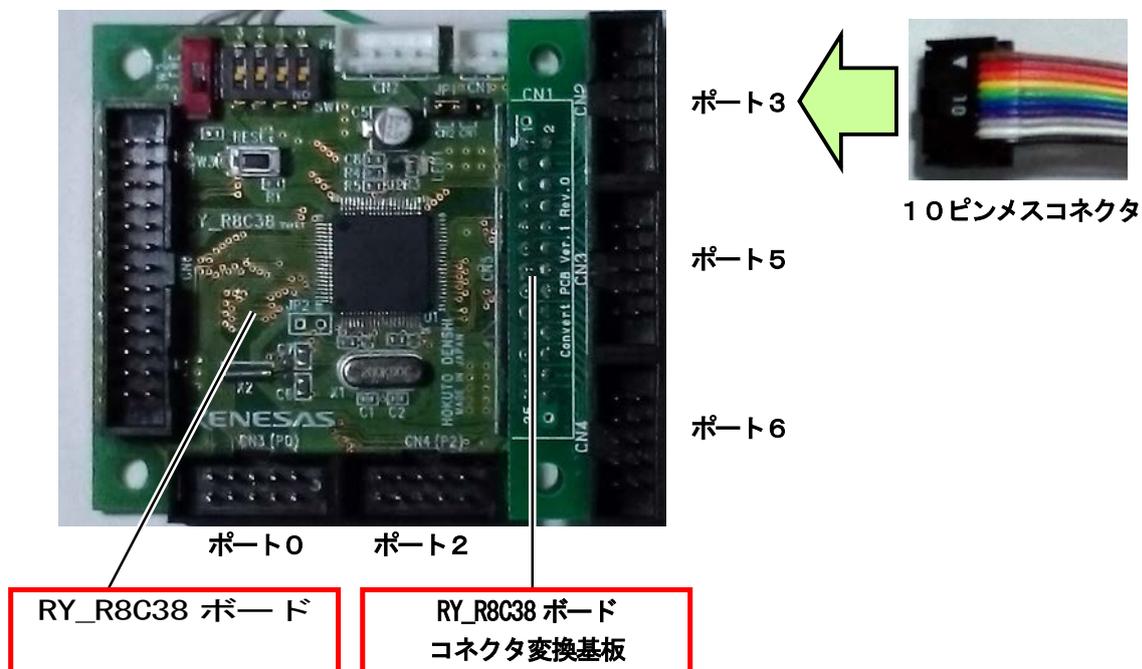


▲ 結線例

3.2 マイコンボードと取り付ける

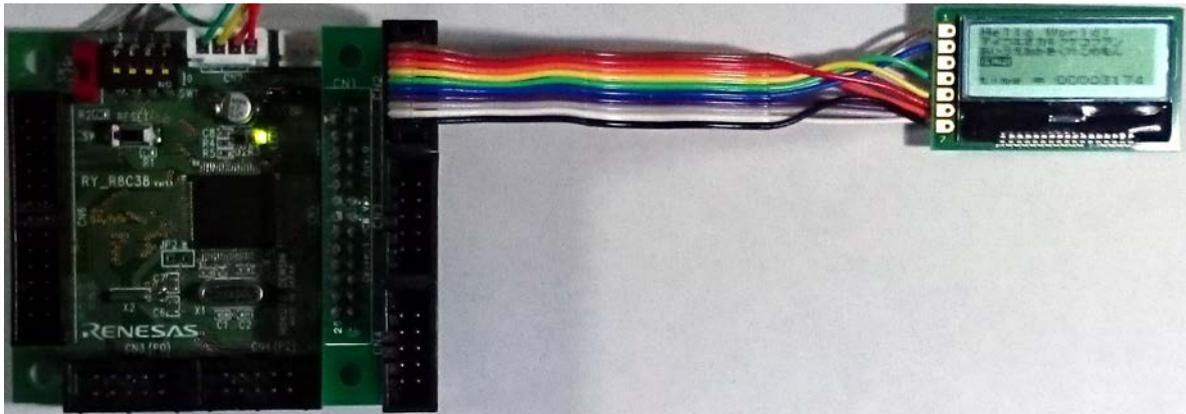
10ピンメスコネクタをポート3に接続します。

RY_R8C38 ボードのポート3は26ピンコネクタで、10ピンメスコネクタを接続することができません。RY_R8C38 ボードコネクタ変換基板を使うと良いでしょう(26ピンメスコネクタを用意して直接、接続しても問題ありません)。



※できる限り、SDI 端子と SCLK 端子の接続ポートは、P3_5 と P3_7 にしてください。プログラムの負荷を軽減することができます。それ以外のポートにも接続可能ですが、プログラムの変更が必要です。RS 端子と CS 端子は、どのポートにも接続することができます。プログラムの変更については「4.3.2

グラフィック液晶と接続するポートの変更」を参照してください。



▲RY_R8C38 ボードに RY_R8C38 ボードコネクタ変換基板を取り付け、グラフィック液晶を接続、表示しているところ

4. プログラム

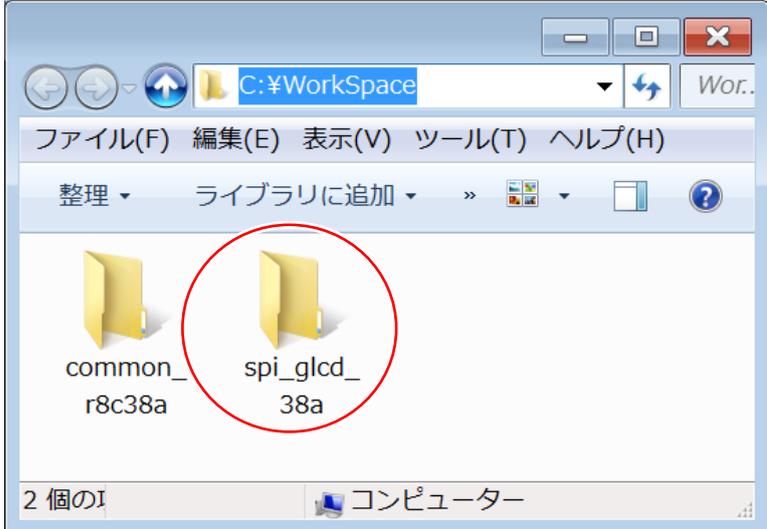
4.1 ワークスペースのインストール

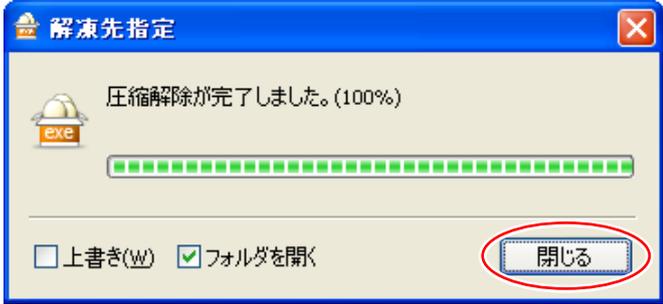
1		<p>マイコンカーラリーホームページ http://www.mcr.gr.jp/index2.html にアクセスします。 「技術情報→ダウンロード」をクリックします。</p>
---	--	--

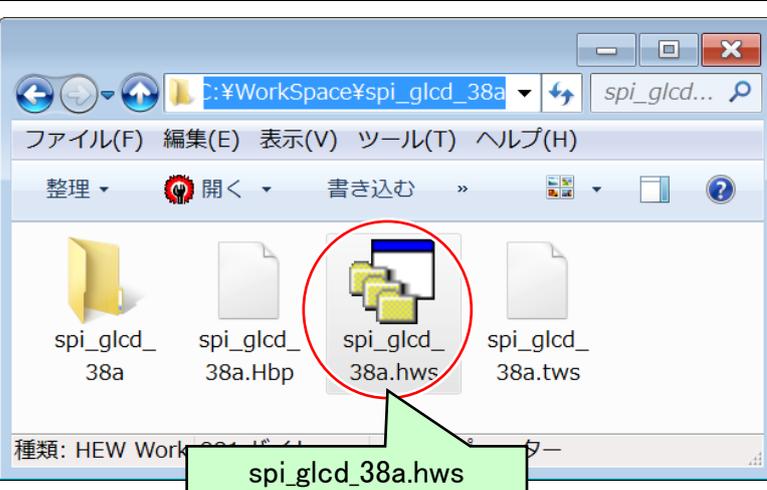
2	<p>免責事項</p> <p>「マニュアル」、「ソフトウェア」は万全な体制で制作されており、通常の使用環境においては正常に動作するように作成されていますが、万が一「マニュアル」、「ソフトウェア」による損失・損害が発生した時には、『ジャパンマイコンカーラリー実行委員会』はいかなる場合も責任を負いません。個人の免責が取れる範囲内であらかじめ了承した上でご使用くださるようお願いをいたします。</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">対象マイコン</th> <th style="width: 55%;">内容</th> <th style="width: 30%;">更新日</th> </tr> </thead> <tbody> <tr> <td>R8C/38A</td> <td>R8C/38Aマイコン(RY_R8C38ボード)に関する資料</td> <td>2014.01.04</td> </tr> <tr> <td>H8/3048F-ONE</td> <td>H8/3048F-ONEマイコン(RY3048Foneボード)用のサンプルプログラム、書き込みソフト</td> <td>2010.10.07</td> </tr> <tr> <td>H8/3048F</td> <td>H8/3048F-ONEマイコン(RY3048Foneボード)に</td> <td>2000.05.25</td> </tr> </tbody> </table>	対象マイコン	内容	更新日	R8C/38A	R8C/38Aマイコン(RY_R8C38ボード)に関する資料	2014.01.04	H8/3048F-ONE	H8/3048F-ONEマイコン(RY3048Foneボード)用のサンプルプログラム、書き込みソフト	2010.10.07	H8/3048F	H8/3048F-ONEマイコン(RY3048Foneボード)に	2000.05.25	<p>「R8C/38A マイコン (RY_R8C38 ボード)に関する資料」をクリックします。</p>
対象マイコン	内容	更新日												
R8C/38A	R8C/38Aマイコン(RY_R8C38ボード)に関する資料	2014.01.04												
H8/3048F-ONE	H8/3048F-ONEマイコン(RY3048Foneボード)用のサンプルプログラム、書き込みソフト	2010.10.07												
H8/3048F	H8/3048F-ONEマイコン(RY3048Foneボード)に	2000.05.25												

3	<p>■その他資料</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 25%;">基板</th> <th style="width: 20%;">製作マニュアル</th> <th style="width: 25%;">プログラム解説マニュアル</th> <th style="width: 30%;">プログラム</th> </tr> </thead> <tbody> <tr> <td>超小型グラフィックLCDピッチ変換 シリアル通信でグラフィック液晶を表示させることのできる基板を制御します。</td> <td>プログラム解説マニュアルを参照してください。</td> <td>SPI接続超小型グラフィック液晶(ST7565R使用)プログラム解説マニュアル 第1.00版 2014.04.28</td> <td style="text-align: center;">spi_glcd_38a.exe 2014.04.28</td> </tr> <tr> <td>12C接続小型LCDモジュール</td> <td></td> <td>12C液晶(ST7032)使用プログラ</td> <td></td> </tr> </tbody> </table>	基板	製作マニュアル	プログラム解説マニュアル	プログラム	超小型グラフィックLCDピッチ変換 シリアル通信でグラフィック液晶を表示させることのできる基板を制御します。	プログラム解説マニュアルを参照してください。	SPI接続超小型グラフィック液晶(ST7565R使用)プログラム解説マニュアル 第1.00版 2014.04.28	spi_glcd_38a.exe 2014.04.28	12C接続小型LCDモジュール		12C液晶(ST7032)使用プログラ		<p>「spi_glcd_38a.exe」をダウンロードします。</p>
基板	製作マニュアル	プログラム解説マニュアル	プログラム											
超小型グラフィックLCDピッチ変換 シリアル通信でグラフィック液晶を表示させることのできる基板を制御します。	プログラム解説マニュアルを参照してください。	SPI接続超小型グラフィック液晶(ST7565R使用)プログラム解説マニュアル 第1.00版 2014.04.28	spi_glcd_38a.exe 2014.04.28											
12C接続小型LCDモジュール		12C液晶(ST7032)使用プログラ												

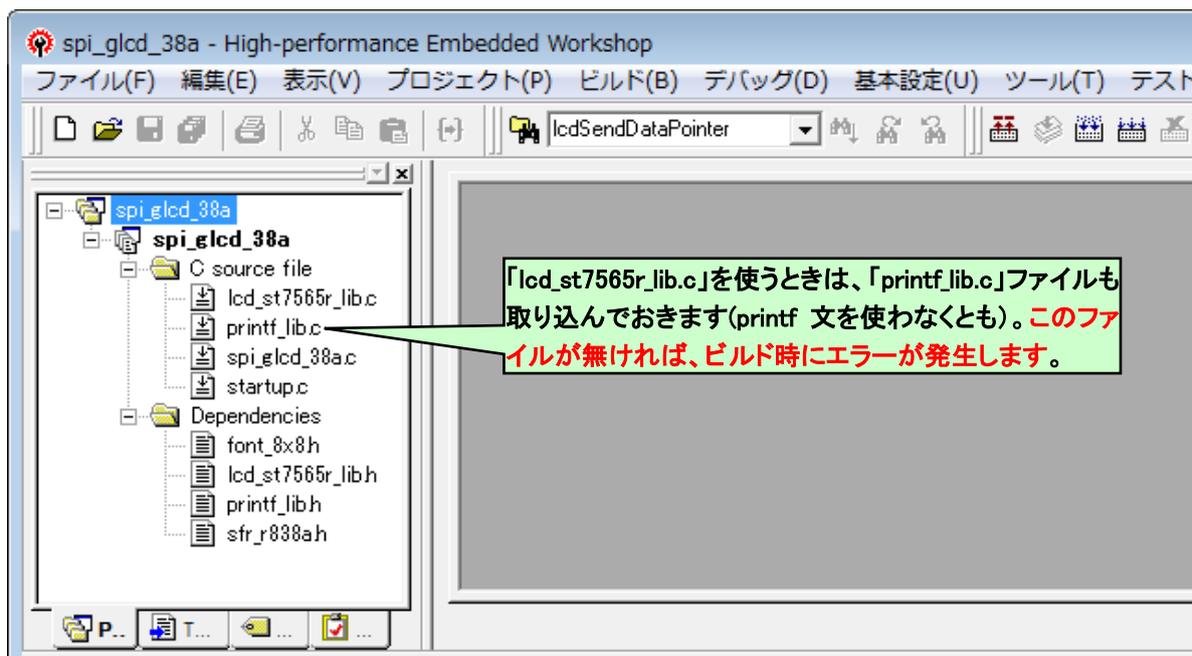
4		<p>圧縮解除をクリックします。</p> <p>※フォルダは変更できません。変更した場合は、ルネサス統合開発環境の設定を変更する場合があります。</p>
---	---	--

5		<p>解凍が終わったら、自動的に「Cドライブ→Workspace」フォルダが開かれます。今回使用するのは、「spi_glcd_38a」です。</p>
---	---	--

6		<p>閉じるをクリックして終了です。</p>
---	---	------------------------

7		<p>「Cドライブ→Workspace→spi_glcd_38a→spi_glcd_38a.hws」をダブルクリックすると、ルネサス統合開発環境が立ち上がります。</p>
---	--	---

4.2 プロジェクト



ワークスペース「spi_glcd_38a」には、1つのプロジェクトが登録されています。

プロジェクト	プログラム	内容
spi_glcd_38a	startup.c	最初に実行されるプログラムです。マイコン固有の設定などを行います。設定が終わると、main 関数に移動します。 このプログラムを編集することはありません。
	spi_glcd_38a.c	main 関数のあるプログラムです。 このプログラムを編集してください。
	lcd_st7565r_lib.c	グラフィック液晶を表示させるためのライブラリ(よく使う関数を集めたファイル)です。このプログラムを編集することはありません(グラフィック液晶の接続ポートを変更する場合は一部、編集します)。
	printf_lib.c	「lcd_st7565r_lib.c」を使うときは、このファイルも取り込んでおきます。 このファイルが無ければ、ビルド時にエラーが発生します。 このプログラムを編集することはありません。

4.3 「lcd_st7565r_lib.c」ファイル

「lcd_st7565r_lib.c」は、グラフィック液晶に文字を表示する専用の関数が用意されているファイルです(ライブラリファイルといいます)。

「lcd_st7565r_lib.c」は、「C:\¥Workspace¥spi_glcd_38a¥spi_glcd_38a」フォルダにあります。

4.3.1 使用できる関数

■initLcd 関数

書式	int initLcd(void)
内容	グラフィック液晶を初期化します。
引数	なし
戻り値	0:異常 1:正常
使用例	<pre>asm(" fset I "); /* 全体の割り込み許可 */ initLcd(); /* グラフィック液晶の初期化 */</pre> <p>※initLcd 関数は、全体割り込み許可をした後に実行してください。</p>

■lcdPrintf 関数

書式	int lcdPrintf(char *format, ...)																																																																																																																														
内容	グラフィック液晶に文字を表示します。書式は、printf 関数と同じです。表示位置は前回表示された続きか、lcdPosition 関数で指定された位置です。																																																																																																																														
引数	char *format 書式文字列 ... 可変個引数																																																																																																																														
戻り値	正常時:出力した文字列 異常時:負の数																																																																																																																														
使用例	<pre>lcdPrintf("0123456789abcdef"); lcdPrintf("ABC");</pre> <p>実行すると下図のようになります。</p> <table border="1" style="margin-left: 40px;"> <tr> <td></td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> <td>列</td> </tr> <tr> <td>0</td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td><td>f</td> <td></td> </tr> <tr> <td>1</td> <td>A</td><td>B</td><td>C</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td> </tr> <tr> <td>2</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td> </tr> <tr> <td>3</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td> </tr> <tr> <td>4</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td> </tr> <tr> <td>5</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> <td></td> </tr> </table>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	列	0	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f		1	A	B	C															2																		3																		4																		5																	
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	列																																																																																																														
0	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f																																																																																																															
1	A	B	C																																																																																																																												
2																																																																																																																															
3																																																																																																																															
4																																																																																																																															
5																																																																																																																															

■lcdPosition 関数

書式	void lcdPosition(char x ,char y)																																																																																																																										
内容	グラフィック液晶に表示する位置を指定します。																																																																																																																										
引数	char x 列 (0 ~ 15) char y 行 (0 ~ 5)																																																																																																																										
戻り値	なし																																																																																																																										
使用例	<pre> lcdPosition(2, 2); /* 表示する位置を指定 */ lcdPrintf("SPI GLCD"); /* 表示する文字列 */ </pre> <p>実行すると下図のようになります。</p> <table border="1" style="margin-left: 40px;"> <tr> <td></td> <td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td> <td>列</td> </tr> <tr> <td>0</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>1</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>2</td> <td></td><td></td><td>S</td><td>P</td><td>I</td><td></td><td>G</td><td>L</td><td>C</td><td>D</td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>3</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>4</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> <tr> <td>5</td> <td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td> </tr> </table>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	列	0																		1																		2			S	P	I		G	L	C	D							3																	4																	5																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	列																																																																																																										
0																																																																																																																											
1																																																																																																																											
2			S	P	I		G	L	C	D																																																																																																																	
3																																																																																																																											
4																																																																																																																											
5																																																																																																																											

■lcdShowProcess 関数

書式	void lcdShowProcess(void)
内容	グラフィック液晶の表示制御を行います。この関数は割り込み処理などで 1ms ごとに実行してください。
引数	なし
戻り値	なし
使用例	<pre> /* LCD 表示処理用関数(1ms ごとに実行) */ lcdShowProcess (); </pre> <p>グラフィック液晶の表示は、1 文字表示に最大で 10ms の時間がかかります(データシート参照)。16 文字表示しようとする、最大で 160ms かかってしまいます。1 回の表示に 160ms も時間を取られては、他の処理が何もできなくなり大問題です。</p> <p>lcdShowProcess 関数は 1ms ごとに実行して、1ms ごとに少しずつグラフィック液晶表示処理をする関数です。この関数は 1ms ごとに実行してください。今回は、タイマ RB の 1ms ごとの割り込み関数内で実行します。</p> <p>【この関数の仕組み】 本来は、1 文字分の表示データをグラフィック液晶に送った後、10ms 待ってから次の表示データを送らなければいけません。 この関数は 1 文字分の表示データをグラフィック液晶に送った後、すぐに関数を終了します。次にこの関数が実行されても、まだ 1ms しか経っていないので何もせずに関数を終了させます。10 回目、次の 1 文字分のデータを送り、またすぐに関数を終了します。これを繰り返します。</p>

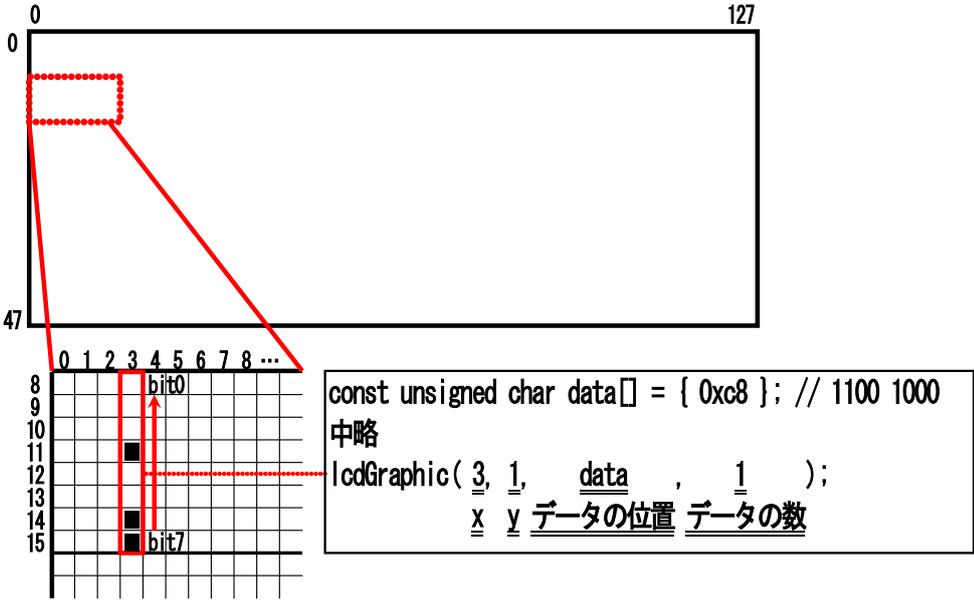
■lcdContrast 関数

書式	void lcdContrast(int i)
内容	グラフィック液晶の表示の濃さを制御します。
引数	0~63 0 が薄く (何も表示されない)、63 が濃く (真っ黒になる) になります。 最初は 60 くらいで表示させて、濃さを実際に確認して調整してください。
戻り値	なし
使用例	lcdContrast(30); /* コントラスト(濃さ)の設定 30 にする */

■lcdKana 関数

書式	void lcdKana(int flag)																																																																																																																							
内容	lcdPrintf 文で、カタカナを表示させるようにしたときに、カタカナを表示させるか、ひらがなを表示させるか設定します。																																																																																																																							
引数	0:変換なし(カタカナ表示) 1:カタカナをひらがなに変換して表示																																																																																																																							
戻り値	なし																																																																																																																							
使用例	<pre> lcdKana(0); // カタカナ表示 lcdPosition(0, 1); lcdPrintf("アイエオカキクワヲン"); lcdKana(1); // ひらがな表示 lcdPosition(0, 2); lcdPrintf("アイエオカキクワヲン"); </pre> <p>実行すると下図のようになります。</p> <table border="1" style="margin-left: 40px;"> <tr> <td></td> <td>0</td> <td>1</td> <td>2</td> <td>3</td> <td>4</td> <td>5</td> <td>6</td> <td>7</td> <td>8</td> <td>9</td> <td>10</td> <td>11</td> <td>12</td> <td>13</td> <td>14</td> <td>15 列</td> </tr> <tr> <td>0</td> <td></td> </tr> <tr> <td>1</td> <td>ア</td> <td>イ</td> <td>ウ</td> <td>エ</td> <td>オ</td> <td>カ</td> <td>キ</td> <td>ク</td> <td>ケ</td> <td>コ</td> <td>ワ</td> <td>ヲ</td> <td>ン</td> <td></td> <td></td> <td></td> </tr> <tr> <td>2</td> <td>あ</td> <td>い</td> <td>う</td> <td>え</td> <td>お</td> <td>か</td> <td>き</td> <td>く</td> <td>け</td> <td>こ</td> <td>わ</td> <td>お</td> <td>ん</td> <td></td> <td></td> <td></td> </tr> <tr> <td>3</td> <td></td> </tr> <tr> <td>4</td> <td></td> </tr> <tr> <td>5</td> <td></td> </tr> </table>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15 列	0																	1	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	ワ	ヲ	ン				2	あ	い	う	え	お	か	き	く	け	こ	わ	お	ん				3																	4																	5																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15 列																																																																																																								
0																																																																																																																								
1	ア	イ	ウ	エ	オ	カ	キ	ク	ケ	コ	ワ	ヲ	ン																																																																																																											
2	あ	い	う	え	お	か	き	く	け	こ	わ	お	ん																																																																																																											
3																																																																																																																								
4																																																																																																																								
5																																																																																																																								

■lcdGraphic 関数

書式	void lcdGraphic(int x, int y, unsigned char _far *adr, int cnt)																																																																																																																																				
内容	液晶にグラフィックデータを表示させます。																																																																																																																																				
引数	int x : 0~127dot int y : 0~5 ※設定値×8 の位置に表示されます unsigned char *adr : 表示させたいデータのアドレス int cnt : 表示させたいデータ数 (x のドット数分)																																																																																																																																				
戻り値	なし																																																																																																																																				
表示位置	<p>横は、0~127 を設定し 1ドットずつ設定します。縦は、0~7 を設定し、設定値×8ドット分の位置に設定されます。下記にx=3、y=1 を設定したときの例を示します。</p> 																																																																																																																																				
	<pre>const unsigned char original_data[] = { /* オリジナルデータ */ 0xff, 0x81, 0xff, 0x85, 0x89, 0x89, 0x85, 0xff, 0x81, 0xbd, 0xc3, 0xc3, 0xc3, 0xa5, 0x81, 0xff, 0x8b, 0x8b, 0x9b, 0xab, 0xc5, 0x81, 0xff, 0x00 }; // オリジナルのデータを表示 lcdGraphic(0, 3, original_data, sizeof(original_data));</pre> <p>実行すると下図のようになります。</p> <table border="1" data-bbox="363 1713 1173 1921"> <tr><td></td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td><td>列</td></tr> <tr><td>0</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>1</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>2</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>3</td><td></td><td></td><td></td><td>M</td><td>C</td><td>R</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>4</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td>5</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	列	0																			1																			2																			3				M	C	R													4																			5																		
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	列																																																																																																																				
0																																																																																																																																					
1																																																																																																																																					
2																																																																																																																																					
3				M	C	R																																																																																																																															
4																																																																																																																																					
5																																																																																																																																					

4.3.2 グラフィック液晶と接続するポートの変更

■マイコンの P3_7 端子、P3_5 端子を使う場合(推奨)

マイコンの P3_7 端子とグラフィック液晶の SDI 端子、マイコンの P3_5 端子とグラフィック液晶の SCLK 端子を接続する場合は、下記のように 22 行目の数字を「1」にします(初期設定値は「1」です)。その他の設定はありません。

```
22 : #define SSU_SFR          1          // 1:マイコンのSFR使用
23 :                          // 0:使用せずにポートで処理
```

■マイコンの P3_7 端子、P3_5 端子を使わない場合

22 行を「0」にします。また、41～44 行目の端子の設定を、接続する端子に合わせて設定します。今回は例として、グラフィック液晶の SDI 端子は P9_0 端子、SCLK 端子は P9_1 端子に接続することとします。

※グラフィック液晶をポート 0 に接続することはできません(PD0 レジスタがプロテクトされているため)。

```
22 : #define SSU_SFR          0          // 1:マイコンのSFR使用
23 :                          // 0:使用せずにポートで処理

中略

40 : // SSU_SFR = 0 のときのSDI端子、SCLK端子の設定 ※ポート0は設定できません
41 : #define lcd_sdi          p9_0       // 液晶のSDI端子
42 : #define lcd_sdiddr      pd9_0      // 液晶のSDI端子の入出力設定レジスタ
43 : #define lcd_sclk        p9_1       // 液晶のSCLK端子
44 : #define lcd_sclkddr     pd9_2      // 液晶のSCLK端子の入出力設定レジスタ
```

■グラフィック液晶の RS 端子、CS 端子の設定

グラフィック液晶の RS 端子、CS 端子をどのポートに接続するか設定します。「SSU_SFR」が 0 でも 1 でも、設定することができます。

35～38 行目の端子の設定を、接続する端子に合わせて設定します。今回は例として、グラフィック液晶の RS 端子は P9_2 端子、CS 端子は P9_3 端子に接続することとします。

※グラフィック液晶をポート 0 に接続することはできません(PD0 レジスタがプロテクトされているため)。

```
33 : // 液晶のRS端子、CS端子は、SSU_SFRが1でも0でも変更できます。
34 : // ※ポート0は設定できません
35 : #define lcd_rs          p9_2       // 液晶のRS端子
36 : #define lcd_rsddr      pd9_2      // 液晶のRS端子の入出力設定レジスタ
37 : #define lcd_cs          p9_3       // 液晶のCS端子
38 : #define lcd_csddr     pd9_3      // 液晶のCS端子の入出力設定レジスタ
```

4.4 「spi_lcd_38a.c」ファイル

4.4.1 プログラム

```

1 : /*******/
2 : /* 対象マイコン R8C/38A or R8C/38C */
3 : /* ファイル内容 グラフィック液晶(ST7565R) 制御プログラム */
4 : /* バージョン Ver. 1.00 */
5 : /* Date 2014.03.23 */
6 : /* Copyright ルネサスマイコンカーラーリ事務局 */
7 : /*******/
8 :
9 : /*=====*/
10 : /* インクルード */
11 : /*=====*/
12 : #include <stdio.h>
13 : #include "sfr_r838a.h" /* R8C/38A SFRの定義ファイル */
14 : #include "lcd_st7565r_lib.h" /* ST7565R液晶表示ライブラリ */
15 : /* 【注意】液晶を使うときは、ルネサス統合開発環境の「C source file」欄に
16 : 「printf_lib.c」も追加してください。無い場合、ビルド時にエラーになります */
17 :
18 : /*=====*/
19 : /* シンボル定義 */
20 : /*=====*/
21 :
22 : /*=====*/
23 : /* プロトタイプ宣言 */
24 : /*=====*/
25 : void init( void );
26 :
27 : /*=====*/
28 : /* グローバル変数の宣言 */
29 : /*=====*/
30 : unsigned long cnt1; /* 1msごとに+1 */
31 :
32 : const unsigned char original_data[] = { /* オリジナルデータ */
33 : 0xff, 0x81, 0xff, 0x85, 0x89, 0x89, 0x85, 0xff,
34 : 0x81, 0xbd, 0xc3, 0xc3, 0xc3, 0xa5, 0x81, 0xff,
35 : 0x8b, 0x8b, 0x9b, 0xab, 0xc5, 0x81, 0xff, 0x00 };
36 :
37 : /*******/
38 : /* メインプログラム */
39 : /*******/
40 : void main( void )
41 : {
42 : /* マイコン機能の初期化 */
43 : init(); /* マイコン内蔵周辺機能の初期化 */
44 : asm(" fset I "); /* 全体の割り込み許可 */
45 : initLcd(); /* 液晶初期化 */
46 :
47 : lcdContrast( 27 ); /* コントラスト(濃さ)の設定 0~63 */
48 :
49 : lcdPosition( 0, 0 );
50 : lcdPrintf( "Hello World!" );
51 :
52 : lcdKana( 0 ); // カタカナ表示
53 : lcdPosition( 0, 1 );
54 : lcdPrintf( "アイエオカキケコサシ" );
55 :
56 : lcdKana( 1 ); // ひらがな表示
57 : lcdPosition( 0, 2 );
58 : lcdPrintf( "アイエオカキケコサシ" );
59 :
60 : // オリジナルのデータを表示
61 : // x(0-127),y(0-5), データのアドレス , データ数
62 : lcdGraphic( 0 , 3 , original_data, sizeof(original_data) );
63 :
64 : lcdPosition( 0, 5 );
65 : lcdPrintf( "time = " );
66 :
67 : while( 1 ) {
68 : lcdPosition( 7, 5 );
69 : lcdPrintf( "%08ld", cnt1 );
70 :
71 : p4_5 = (cnt1 / 500 ) & 0x1; /* マイコンボードのLEDを点滅 */
72 : }

```

```

73 : }
74 :
75 : /*****
76 : /* R8C/38A スペシャルファンクションレジスタ (SFR) の初期化 */
77 : /*****
78 : void init( void )
79 : {
80 :     int i;
81 :
82 :     /* クロックをXINクロック (20MHz)に変更 */
83 :     prc0 = 1; /* プロテクト解除 */
84 :     cm13 = 1; /* P4_6, P4_7をXIN-XOUT端子にする*/
85 :     cm05 = 0; /* XINクロック発振 */
86 :     for(i=0; i<50; i++ ); /* 安定するまで少し待つ(約10ms) */
87 :     ocd2 = 0; /* システムクロックをXINにする */
88 :     prc0 = 0; /* プロテクトON */
89 :
90 :     /* ポートの入出力設定 */
91 :     prc2 = 1; /* PD0のプロテクト解除 */
92 :     pd0 = 0x00; /*
93 :     pd1 = 0xd0; /* 5:RXD0 4:TXD0 3-0:DIP SW */
94 :     pd2 = 0x00; /*
95 :     pd3 = 0x00; /*
96 :     p4 = 0x20; /* P4_5のLED:初期は点灯 */
97 :     pd4 = 0xb8; /* 7:XOUT 6:XIN 5:LED 2:VREF */
98 :     pd5 = 0x00; /*
99 :     pd6 = 0x00; /*
100 :     pd7 = 0x00; /*
101 :     pd8 = 0x00; /*
102 :     pd9 = 0x00; /*
103 :     pur0 = 0x04; /* P1_3~P1_0のプルアップON */
104 :
105 :     /* タイマRBの設定 */
106 :     /* 割り込み周期 = 1 / 20[MHz] * (TRBPRE+1) * (TRBPR+1)
107 :     = 1 / (20*106) * 200 * 100
108 :     = 0.001[s] = 1[ms]
109 :     */
110 :     trbmr = 0x00; /* 動作モード、分周比設定 */
111 :     trbpre = 200-1; /* プリスケアラレジスタ */
112 :     trbpr = 100-1; /* プライマリレジスタ */
113 :     trbic = 0x07; /* 割り込み優先レベル設定 */
114 :     trbcr = 0x01; /* カウント開始 */
115 : }
116 :
117 : /*****
118 : /* タイマRB 割り込み処理 */
119 : /*****
120 : #pragma interrupt intTRB(vect=24)
121 : void intTRB( void )
122 : {
123 :     cnt1++;
124 :
125 :     /* 液晶表示処理用関数(1msごとに実行) */
126 :     lcdShowProcess();
127 : }
128 :
129 : /*****
130 : /* end of file */
131 : /*****

```

グラフィック液晶が繋がつている端子は、入力でも出力でも、どちらでも構いません(initLcd 関数で設定するため)。

4.4.2 プログラムの解説

(1) ヘッダファイルのインクルード

```

 9 : /*=====*/
10 : /* インクルード */
11 : /*=====*/
12 : #include <stdio.h>
13 : #include "sfr_r838a.h" /* R8C/38A SFRの定義ファイル */
14 : #include "lcd_st7565r_lib.h" /* ST7565R液晶表示ライブラリ */

```

14 行	グラフィック液晶制御ライブラリのヘッダファイルをインクルードします。これで、「lcd_st7565r_lib.c」内の関数を使うことができます。
------	--

(2) グラフィック液晶の初期化、コントラストの設定

```

37 : /*=====*/
38 : /* メインプログラム */
39 : /*=====*/
40 : void main( void )
41 : {
42 :     /* マイコン機能の初期化 */
43 :     init(); /* マイコン内蔵周辺機能の初期化 */
44 :     asm(" fset I "); /* 全体の割り込み許可 */
45 :     initLcd(); /* 液晶初期化 */
46 :
47 :     lcdContrast( 27 ); /* コントラスト(濃さ)の設定 0~63 */

```

45 行	グラフィック液晶の初期化をします。initLcd 関数は、全体の割り込みが許可された状態（「asm(" fset I ")」以降）で実行してください。
47 行	グラフィック液晶のコントラスト(表示の濃さ)の調整をします。値は 0~63 を設定します。0 が薄く(何も表示されない)、63 が濃く(真っ黒になる)なります。最初は、30 くらいで表示させて、濃さを実際に確認して調整してください。

(3) グラフィック液晶に表示する

```

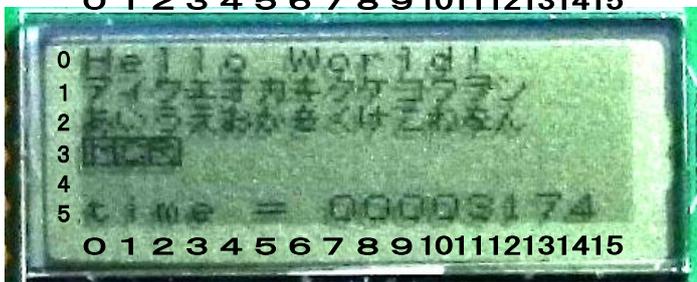
49 :     lcdPosition( 0, 0 );
50 :     lcdPrintf( "Hello World!" );
51 :
52 :     lcdKana( 0 ); // カタカナ表示
53 :     lcdPosition( 0, 1 );
54 :     lcdPrintf( "アイウエオカキククワソ" );
55 :
56 :     lcdKana( 1 ); // ひらがな表示
57 :     lcdPosition( 0, 2 );
58 :     lcdPrintf( "アイウエオカキククワソ" );
59 :
60 :     // オリジナルのデータを表示

```

```

61 : //          x(0-127),y(0-5), データのアドレス , データ数
62 : lcdGraphic( 0          ,3          , original_data, sizeof(original_data) );
63 :
64 : lcdPosition( 0, 5 );
65 : lcdPrintf( "time = " );
66 :
67 : while( 1 ) {
68 :     lcdPosition( 7, 5 );
69 :     lcdPrintf( "%08ld", cnt1 );
70 :
71 :     p4_5 = (cnt1 / 500 ) & 0x1;    /* マイコンボードのLEDを点滅    */
72 : }

```

<p>49 行～ 69 行</p>	<p>下記のように表示されます。</p>  <p>5 行目には、1ms ごとにカウントアップする cnt1 変数の値が表示されます。</p>
<p>71 行</p>	<p>グラフィック液晶とは関係ないですが、マイコンが動作していると分かるように、マイコンボード上の LED を 500ms ごとに点灯、消灯させています。</p>

(4) グラフィック液晶表示処理関数

```

117 : /*****
118 : /* タイマRB 割り込み処理
119 : *****/
120 : #pragma interrupt intTRB(vect=24)
121 : void intTRB( void )
122 : {
123 :     cnt1++;
124 :
125 :     /* 液晶表示処理用関数(1msごとに実行)
126 :     lcdShowProcess();
127 : }

```

<p>126 行</p>	<p>lcdPosition 関数や lcdPrintf 関数は、表示をする準備をしているだけです。実際に表示させているのが lcdShowProcess 関数 です。1ms ごとにこの関数を実行します。今回は、タイマ RB 割り込み関数が 1ms ごとに実行されているので、この関数内に lcdShowProcess 関数を記述しています。</p>
--------------	--

5. 参考文献

- ルネサス エレクトロニクス(株)
R8C/38C グループ ユーザーズマニュアル ハードウェア編 Rev.1.10
- ルネサス エレクトロニクス(株)
M16C シリーズ,R8C ファミリー用 C/C++コンパイラパッケージ V.6.00
C/C++コンパイラユーザーズマニュアル Rev.1.00
- ルネサス エレクトロニクス(株)
High-performance Embedded Workshop V.4.09 ユーザーズマニュアル Rev.1.00
- ルネサス半導体トレーニングセンター C言語入門コーステキスト 第1版
- 電波新聞社 マイコン入門講座 大須賀威彦著 第1版
- ソフトバンク(株) 新C言語入門シニア編 林晴比古著 初版
- 共立出版(株) プログラマのための ANSI C 全書 L.Ammeraal 著
吉田敬一・竹内淑子・吉田恵美子訳 初版

マイコンカーラリー、販売部品についての詳しい情報は、マイコンカーラリー販売サイトをご覧ください。

<https://www2.himdx.net/mcr/>

R8C マイコンについての詳しい情報は、ルネサス エレクトロニクスのホームページをご覧ください。

<http://japan.renesas.com/>

の「製品情報」欄→「マイコン」→「R8C」でご覧頂けます